9장. 타입 변환과 단축 평가

9장. 타입 변환과 단축 평가

9.1 타입 변환이란?

1. **명시적 타입 변환**(explicit coercion), **타입 캐스팅**(type casting)

   : 개발자가 의도적으로 값의 타입을 변환하는 것

   [예제 09 - 01]

```
var x = 10;

// 명시적 타입 변환
// 숫자를 문자열로 타입 캐스팅한다.
var str = x.toString();
console.log(typeof str, str);   // string 10

// x 변수의 값이 변경된 것은 아니다.
console.log(typeof x, x);       // number 10
```

2. **암묵적 타입 변환**(implicit coercion), **타입 강제 변환**(type coercion)

   : 개발자 의도와 상관없이, 자바스크립트 엔진에 의해 암묵적으로 타입이 자동 변환

9.2 암묵적 타입 변환 (개발자 의도 X)

1. [예제 09 - 03]

```
'10' + 2  // '102'
```

```
5 * '10'  //  50
```

- 자바스크립트는 가급적 에러를 발생시키지 않도록 **암묵적 타입 변환**을 통해 표현식 평가
- 암묵적 타입 변환 : 문자열, 숫자, 불리언과 같은 원시 타입 중 하나로 자동 변환

9.2.1 문자열 타입으로 변환

1. [예제 09 - 04]

```
1 + '2' // "12"
```

- 자바스크립트 엔진은 문자열 연결 연산자 표현식을 평가하기 위해 문자열 연결 연산자의 피연산자 중에서 문자열 타입이 아닌 피연산자를 문자열 타입으로 암묵적 타입 변환

9.2.2 숫자 타입으로 변환

9.2.3 불리언 타입으로 변환

1. [예제 09 - 11]

```
if('') console.log('1');
if(true) console.log('2');
if(0) console.log('3');
if('str') console.log('4');
if(null) console.log('5');

// 2 4
```

- **자바스크립트 엔진은 Boolean 타입이 아닌 값을**

  Truthy(참으로 평가되는 값) or **Falsy 값(거짓으로 평가되는 값)**으로 구분

2. false로 평가되는 Falsy 값들 예시

- false
- undefined
- null
- 0, -0
- NaN
- ' ' (빈 문자열)

9.3 명시적 타입 변환 (개발자 의도 O)

1. 표준 빌트인 생성자 함수, 표준 빌트인 메서드 : 자바스크립트에서 기본 제공하는 함수
    - new 연산자와 호출 /

9.3.1 문자열 타입으로 변환

1. 문자열 타입이 아닌 값 → 문자열 타입으로 변환하는 방법 3가지
    a. String 생성자 함수를 new 연산자 없이 호출하는 방법
    b. Object.prototype.toString 메서드를 사용하는 방법
    c. 문자열 연결 연산자를 이용하는 방법

9.3.2 숫자 타입으로 변환

9.3.2 숫자 타입으로 변환

9.4 단축 평가

9.4.1 논리 연산자를 사용한 단축 평가

1. 논리합(||) 또는 논리곱(&&) 연산자 표현식은 2개의 피연산자 중 어느 한쪽으로 평가된다.

2. [예제 09 - 17]

```
'Cat' && 'Dog'  // "Dog"
```

- **논리 연산의 결과를 결정하는 두 번째 피연산자, 즉 문자열 'Dog'를 그대로 반환**

3. [예제 09 - 18]

```
'Cat' || 'Dog'  // "Cat"
```

- **논리 연산의 결과를 결정하는 첫 번째 피연산자, 즉 문자열 'Cat'를 그대로 반환**

4. **단축 평가**(short-circuit evaluation)

   : **논리 연산의 결과를 결정하는 피연산자를 타입 변환하지 않고 그대로 반환**

   - **표현식을 평가하는 도중, 평가 결과가 확정된 경우 나머지 평가 과정을 생략하는 것을 의미**

9.4.2 옵셔널 체이닝 연산자

1. 옵셔널 체이닝(optional chaining) 연산자 ?.
   : 좌항의 피연산자가 null or undefined 일 경우 undefined 반환, 아니면 우항의 프로퍼티 참조

2. 옵셔널 체이닝 연산자 ?. 는,

   객체 가리키기를 기대하는 변수가 null, undefined 가 아닌지 확인하고 프로퍼티를 참조할 때 유용

9.4.3 null 병합 연산자

1. [예제 09 - 30]

```
var foo = null ?? 'default string';
console.log(foo); // "default string"
```

2. [예제 09 - 32]

```
var foo = '' ?? 'default string';
console.log(foo); // ""
```